

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT: Matula et al.

SERIAL NO.: (not yet assigned)

FILING DATE: May 2, 2001

TITLE: APPARATUS FOR DYNAMIC IMPLEMENTATION OF
JAVA™ METADATA INTERFACES

EXAMINER: [not yet assigned]

ART UNIT: [not yet assigned]

CERTIFICATE OF MAILING

"Express Mail" mailing label no.: EL839725974US

Date of Deposit: May 2, 2001

I hereby certify that this paper is being deposited with the United States Postal Service "Express Mail

Post Office to Addressee" service under 37 CFR §1.10 on the date indicated above and addressed to:

Box PATENT APPLICATION, Commissioner for Patents, Washington, DC 20231, on the date printed below:

Name: Diane Morse
Diane Morse

**BOX PATENT APPLICATION
COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231**

PRELIMINARY AMENDMENT

Please amend the above-identified application as follows.

In the Title:

Please change "METHOD FOR DYNAMIC IMPLEMENTATION OF JAVA™
METADATA INTERFACES" to --APPARATUS FOR DYNAMIC IMPLEMENTATION OF
JAVA™ METADATA INTERFACES--.

In the Claims:

Please cancel claims 2-66 without prejudice.

Please add new claims 67-122 as follows.

67. (NEW) An apparatus for dynamic implementation of a Java™ Metadata Interface (JMI) to a metamodel, the apparatus comprising:

means for receiving a JMI implementation request, said request associated with a metamodel, said metamodel comprising at least one package, said at least one package comprising at least one class, said at least one class comprising at least one attribute, reference or operation;

means for implementing a package proxy JMI interface when said request comprises a package proxy request;

means for implementing a class proxy JMI interface when said request comprises a class proxy request; and

means for implementing a class instance JMI interface when said request comprises a class instance request.

68. (NEW) The apparatus of claim 67 wherein said means for implementing a package proxy JMI interface comprises:

means for generating bytecode for a class that implements said package proxy JMI interface;

means for creating a new instance of said class; and

means for returning said instance.

69. (NEW) The apparatus of claim 68 wherein said means for generating further comprises:

means for receiving a metamodel package;

means for receiving a package proxy interface method associated with said metamodel package;

means for determining a class name based upon said interface method;
means for searching said metamodel package for a class corresponding to said class name;
and
means for producing an implementation of said interface method that returns the proxy for
said class when said class name is found in said package.

70. (NEW) The apparatus of claim 69 wherein said implementation of said interface method
calls a handler method of the superclass of said class, passing said class name as an argument
and returning the proxy for said class.

71. (NEW) The apparatus of claim 67 wherein said means for implementing a class proxy JMI
interface comprises:

means for generating bytecode for a class that implements said class proxy JMI interface;
means for creating a new instance of said class; and
means for returning said instance.

72. (NEW) The apparatus of claim 71 wherein said means for generating further comprises:

means for receiving a metamodel class;
means for receiving a class proxy interface method associated with said metamodel class;
means for producing a first implementation of said interface method that creates a new
instance of said class when said method is parameterless; and

means for producing a second implementation of said interface method that creates a new instance of said class and sets the attributes passed as arguments to said interface method when said interface method includes at least one parameter.

73. (NEW) The apparatus of claim 72 wherein said first implementation calls a handler method of the superclass of said class, passing said class name as an argument and returning a new instance of said class.

74. (NEW) The apparatus of claim 72 wherein said second implementation calls a handler method of the superclass of said class, passing said class name, attributes and attribute values as arguments and returning a new instance of said class.

75. (NEW) The apparatus of claim 67 wherein said means for implementing a class instance JMI interface comprises:

means for generating bytecode for a class that implements said class instance JMI interface; means for creating a new instance of said class; and means for returning said instance.

76. (NEW) The apparatus of claim 75 wherein said means for generating further comprises:

means for receiving a metamodel class; means for receiving a class instance interface method associated with said metamodel class, said interface method having an interface method name;

means for producing a first implementation of said interface method that sets the value of an attribute when said interface method name includes a first prefix and when the attribute associated with said interface method is found in said metamodel class;

means for producing a second implementation of said interface method that sets the value of a reference when said interface method name includes a first prefix and when the reference associated with said interface method is found in said metamodel class;

means for producing a third implementation of said interface method that gets the value of an attribute when said interface method name includes a second prefix and when the attribute associated with said interface method is found in said metamodel class;

means for producing a fourth implementation of said interface method that gets the value of a reference when said interface method name includes a second prefix and when the reference associated with said interface method is found in said metamodel class; and

means for producing a fifth implementation of said interface method that executes an operation when said interface method has the same name as said operation.

77. (NEW) The apparatus of claim 76 wherein

said first prefix is “set”; and

said second prefix is “get”.

78. (NEW) The apparatus of claim 76 wherein said means for producing a first implementation further comprises:

means for receiving an attribute name and an attribute value; and

means for producing an implementation that calls a handler method of the superclass of said class, passing said attribute name and attribute value as arguments.

79. (NEW) The apparatus of claim 76 wherein said means for producing a second implementation further comprises:

means for receiving a reference name and an reference value; and

means for producing an implementation that calls a handler method of the superclass of said class, passing said reference name and reference value as arguments.

80. (NEW) The apparatus of claim 76 wherein said means for producing a third implementation further comprises:

means for receiving an attribute name;

means for producing an implementation that calls a handler method of the superclass of said class, passing said attribute name as an argument and returning the attribute value associated with said attribute name; and

means for returning said attribute value.

81. (NEW) The apparatus of claim 76 wherein said means for producing a fourth implementation further comprises:

means for receiving a reference name;

means for producing an implementation that calls a handler method of the superclass of said class, passing said reference name as an argument and returning the reference value associated with said reference name; and

means for returning said reference value.

82. (NEW) The apparatus of claim 76 wherein said means for producing a fifth implementation further comprises:

means for receiving an operation name and any associated arguments;

means for producing an implementation that calls a handler method of the superclass of said class, passing said operation name and said associated arguments as arguments and returning an operation return value; and

means for returning said operation return value.

83. (NEW) An apparatus for dynamic implementation of a Java™ Metadata Interface (JMI), the apparatus comprising:

means for receiving a JMI implementation request, said request associated with a metamodel, said metamodel comprising at least one package, said at least one package comprising at least one class, said at least one class comprising at least one attribute, reference or operation;

means for implementing a JMI interface when said JMI interface is unimplemented; and

means for executing a stored JMI interface implementation when said JMI interface is implemented.

84. (NEW) The apparatus of claim 83 wherein

said means for implementing further comprises:

means for implementing a package proxy JMI interface when said request comprises a package proxy request and when said package proxy JMI interface is unimplemented;

means for implementing a class proxy JMI interface when said request comprises a class proxy request and when said class proxy JMI interface is unimplemented; and

means for implementing a class instance JMI interface when said request comprises a class instance request and when said class instance JMI interface is unimplemented;

and

said means for executing further comprises:

means for executing a stored a package proxy JMI interface implementation when said request comprises a package proxy request and when said package proxy JMI interface is implemented;

means for executing a stored class proxy JMI interface when said request comprises a class proxy request and when said class proxy JMI interface is implemented; and

means for executing a stored class instance JMI interface when said request comprises a class instance request and when said class instance JMI interface is implemented.

85. (NEW) The apparatus of claim 84 wherein said means for implementing a package proxy JMI interface comprises:

means for generating bytecode for a class that implements said package proxy JMI interface;

means for creating a new instance of said class; and

means for returning said instance.

86. (NEW) The apparatus of claim 85 wherein said means for generating further comprises:

means for receiving a metamodel package;

means for receiving a package proxy interface method associated with said metamodel package;

means for determining a class name based upon said interface method;
means for searching said metamodel package for a class corresponding to said class name;
and
means for producing an implementation of said interface method that returns the proxy for
said class when said class name is found in said package.

87. (NEW) The apparatus of claim 86 wherein said implementation of said interface method
calls a handler method of the superclass of said class, passing said class name as an argument
and returning the proxy for said class.

88. (NEW) The apparatus of claim 84 wherein said means for implementing a class proxy JMI
interface comprises:

means for generating bytecode for a class that implements said class proxy JMI interface;
means for creating a new instance of said class; and
means for returning said instance.

89. (NEW) The apparatus of claim 88 wherein said generating further comprises:

means for receiving a metamodel class;
means for receiving a class proxy interface method associated with said metamodel class;
means for producing a first implementation of said interface method that creates a new
instance of said class when said method is parameterless; and

means for producing a second implementation of said interface method that creates a new instance of said class and sets the attributes passed as arguments to said interface method when said interface method includes at least one parameter.

90. (NEW) The apparatus of claim 89 wherein said first implementation calls a handler method of the superclass of said class, passing said class name as an argument and returning a new instance of said class.

91. (NEW) The apparatus of claim 89 wherein said second implementation calls a handler method of the superclass of said class, passing said class name, attributes and attribute values as arguments and returning a new instance of said class.

92. (NEW) The apparatus of claim 84 wherein said means for implementing a class instance JMI interface comprises:

means for generating bytecode for a class that implements said class instance JMI interface; means for creating a new instance of said class; and means for returning said instance.

93. (NEW) The apparatus of claim 92 wherein said means for generating further comprises:

means for receiving a metamodel class; means for receiving a class instance interface method associated with said metamodel class, said interface method having an interface method name;

means for producing a first implementation of said interface method that sets the value of an attribute when said interface method name includes a first prefix and when the attribute associated with said interface method is found in said metamodel class;

means for producing a second implementation of said interface method that sets the value of a reference when said interface method name includes a first prefix and when the reference associated with said interface method is found in said metamodel class;

means for producing a third implementation of said interface method that gets the value of an attribute when said interface method name includes a second prefix and when the attribute associated with said interface method is found in said metamodel class;

means for producing a fourth implementation of said interface method that gets the value of a reference when said interface method name includes a second prefix and when the reference associated with said interface method is found in said metamodel class; and

means for producing a fifth implementation of said interface method that executes an operation when said interface method has the same name as said operation.

94. (NEW) The apparatus of claim 93 wherein

said first prefix is “set”; and

said second prefix is “get”.

95. (NEW) The apparatus of claim 93 wherein said means for producing a first implementation further comprises:

means for receiving an attribute name and an attribute value; and

means for producing an implementation that calls a handler method of the superclass of said class, passing said attribute name and attribute value as arguments.

96. (NEW) The apparatus of claim 93 wherein said means for producing a second implementation further comprises:

means for receiving a reference name and an reference value; and

means for producing an implementation that calls a handler method of the superclass of said class, passing said reference name and reference value as arguments.

97. (NEW) The apparatus of claim 93 wherein said means for producing a third implementation further comprises:

means for receiving an attribute name;

means for producing an implementation that calls a handler method of the superclass of said class, passing said attribute name as an argument and returning the attribute value associated with said attribute name; and

means for returning said attribute value.

98. (NEW) The apparatus of claim 93 wherein said means for producing a fourth implementation further comprises:

means for receiving a reference name;

means for producing an implementation that calls a handler method of the superclass of said class, passing said reference name as an argument and returning the reference value associated with said reference name; and

means for returning said reference value.

99. (NEW) The apparatus of claim 93 wherein said means for producing a fifth implementation further comprises:

means for receiving an operation name and any associated arguments;
means for producing an implementation that calls a handler method of the superclass of said class, passing said operation name and said associated arguments as arguments and returning an operation return value; and
means for returning said operation return value.

100. (NEW) An apparatus for dynamic implementation of a Java™ Metadata Interface (JMI) to a metamodel, the apparatus comprising:

a requestor to make a JMI implementation request, said request associated with a metamodel, said metamodel comprising at least one package, said at least one package comprising at least one class, said at least one class comprising at least one attribute, reference or operation;
a package proxy implementor to implement a package proxy JMI interface when said request comprises a package proxy request;
a class proxy implementor to implement a class proxy JMI interface when said request comprises a class proxy request; and
a class instance implementor to implement a class instance JMI interface when said request comprises a class instance request.

101. (NEW) The apparatus of claim 100 wherein said package proxy implementor is further configured to:

generate bytecode for a class that implements said package proxy JMI interface;
create a new instance of said class; and
return said instance.

102. (NEW) The apparatus of claim 101 wherein said package proxy implementor is further configured to

receive a metamodel package;
receive a package proxy interface method associated with said metamodel package;
determine a class name based upon said interface method;
search said metamodel package for a class corresponding to said class name; and
produce an implementation of said interface method that returns the proxy for said class when said class name is found in said package.

103. (NEW) The apparatus of claim 102 wherein said implementation of said interface method calls a handler method of the superclass of said class, passing said class name as an argument and returning the proxy for said class.

104. (NEW) The apparatus of claim 100 wherein said class proxy implementor is further configured to:

generate bytecode for a class that implements said class proxy JMI interface;
create a new instance of said class; and
return said instance.

105. (NEW) The apparatus of claim 104 wherein said class proxy implementor is further configured to:

receive a metamodel class;

receive a class proxy interface method associated with said metamodel class;

produce a first implementation of said interface method that creates a new instance of said class when said method is parameterless; and

produce a second implementation of said interface method that creates a new instance of said class and sets the attributes passed as arguments to said interface method when said interface method includes at least one parameter.

106. (NEW) The apparatus of claim 105 wherein said first implementation calls a handler method of the superclass of said class, passing said class name as an argument and returning a new instance of said class.

107. (NEW) The apparatus of claim 105 wherein said second implementation calls a handler method of the superclass of said class, passing said class name, attributes and attribute values as arguments and returning a new instance of said class.

108. (NEW) The apparatus of claim 99 wherein said class instance implementor is further configured to:

generate bytecode for a class that implements said class instance JMI interface;
create a new instance of said class; and
return said instance.

109. (NEW) The apparatus of claim 108 wherein said class instance implementor is further configured to:

receive a metamodel class;
receive a class instance interface method associated with said metamodel class, said interface method having an interface method name;
produce a first implementation of said interface method that sets the value of an attribute when said interface method name includes a first prefix and when the attribute associated with said interface method is found in said metamodel class;
produce a second implementation of said interface method that sets the value of a reference when said interface method name includes a first prefix and when the reference associated with said interface method is found in said metamodel class;
produce a third implementation of said interface method that gets the value of an attribute when said interface method name includes a second prefix and when the attribute associated with said interface method is found in said metamodel class;
produce a fourth implementation of said interface method that gets the value of a reference when said interface method name includes a second prefix and when the reference associated with said interface method is found in said metamodel class; and
produce a fifth implementation of said interface method that executes an operation when said interface method has the same name as said operation.

110. (NEW) The apparatus of claim 109 wherein

 said first prefix is “set”; and

 said second prefix is “get”.

111. (NEW) An apparatus for dynamic implementation of a Java™ Metadata Interface (JMI), the apparatus comprising:

 a requestor to make a JMI implementation request, said request associated with a metamodel,

 said metamodel comprising at least one package, said at least one package comprising at

 least one class, said at least one class comprising at least one attribute, reference or

 operation;

 an implementor to implement a JMI interface when said JMI interface is unimplemented; and

 an executor to execute a stored JMI interface implementation when said JMI interface is

 implemented.

112. (NEW) The apparatus of claim 111 wherein

 said implementor further comprises:

 a package proxy implementor to implement a JMI interface when said request comprises

 a package proxy request and when said package proxy JMI interface is

 unimplemented;

 a class proxy implementor to implement a JMI interface when said request comprises a

 class proxy request and when said class proxy JMI interface is unimplemented; and

a class instance implementor to implement a JMI interface when said request comprises a class instance request and when said class instance JMI interface is unimplemented; and

said executor is further configured to:

execute a stored a package proxy JMI interface implementation when said request comprises a package proxy request and when said package proxy JMI interface is implemented;

execute a stored class proxy JMI interface when said request comprises a class proxy request and when said class proxy JMI interface is implemented; and

execute a stored class instance JMI interface when said request comprises a class instance request and when said class instance JMI interface is implemented.

113. (NEW) The apparatus of claim 112 wherein said package proxy implementor is further configured to:

generate bytecode for a class that implements said package proxy JMI interface; create a new instance of said class; and return said instance.

114. (NEW) The apparatus of claim 113 wherein said package proxy implementor is further configured to:

receive a metamodel package;

receive a package proxy interface method associated with said metamodel package;

determine a class name based upon said interface method;

search said metamodel package for a class corresponding to said class name; and

produce an implementation of said interface method that returns the proxy for said class when said class name is found in said package.

115. (NEW) The apparatus of claim 114 wherein said implementation of said interface method calls a handler method of the superclass of said class, passing said class name as an argument and returning the proxy for said class.

116. (NEW) The apparatus of claim 112 wherein said class proxy implementor is further configured to:

generate bytecode for a class that implements said class proxy JMI interface; create a new instance of said class; and return said instance.

117. (NEW) The apparatus of claim 116 wherein said class proxy implementor is further configured to:

receive a metamodel class; receive a class proxy interface method associated with said metamodel class; produce a first implementation of said interface method that creates a new instance of said class when said method is parameterless; and produce a second implementation of said interface method that creates a new instance of said class and sets the attributes passed as arguments to said interface method when said interface method includes at least one parameter.

118. (NEW) The apparatus of claim 117 wherein said first implementation calls a handler method of the superclass of said class, passing said class name as an argument and returning a new instance of said class.

119. (NEW) The apparatus of claim 117 wherein said second implementation calls a handler method of the superclass of said class, passing said class name, attributes and attribute values as arguments and returning a new instance of said class.

120. (NEW) The apparatus of claim 112 wherein said class instance implementor is further configured to:

- generate bytecode for a class that implements said class instance JMI interface;
- create a new instance of said class; and
- return said instance.

121. (NEW) The apparatus of claim 120 wherein said class instance implementor is further configured to:

- receive a metamodel class;
- receive a class instance interface method associated with said metamodel class, said interface method having an interface method name;
- produce a first implementation of said interface method that sets the value of an attribute when said interface method name includes a first prefix and when the attribute associated with said interface method is found in said metamodel class;

produce a second implementation of said interface method that sets the value of a reference when said interface method name includes a first prefix and when the reference associated with said interface method is found in said metamodel class;

produce a third implementation of said interface method that gets the value of an attribute when said interface method name includes a second prefix and when the attribute associated with said interface method is found in said metamodel class;

produce a fourth implementation of said interface method that gets the value of a reference when said interface method name includes a second prefix and when the reference associated with said interface method is found in said metamodel class; and

produce a fifth implementation of said interface method that executes an operation when said interface method has the same name as said operation.

122. (NEW) The apparatus of claim 121 wherein

 said first prefix is “set”; and
 said second prefix is “get”.

REMARKS

Claims 2-66 have been canceled without prejudice.

Claims 67-122 have been added because they particularly point out and distinctly claim subject matter regarded as the invention.

Request for Allowance

It is believed that this Amendment places the above-identified patent application into condition for allowance. Early favorable consideration of this Amendment is earnestly solicited.

If, in the opinion of the Examiner, an interview would expedite the prosecution of this application, the Examiner is invited to call the undersigned attorney at the number indicated below.

Respectfully submitted,
THELEN REID & PRIEST LLP



Marc S. Hanish
Reg. No. 42,626

Dated: May 2, 2001

THELEN REID & PRIEST LLP
P.O. Box 640640
San Jose, CA 95164-0640
(408) 292-5800

VERSION WITH MARKINGS TO SHOW CHANGES MADEIN THE CLAIMS

Claims 2-66 have been cancelled.

Claims 67-122 have been added as follows:

67. (NEW) An apparatus for dynamic implementation of a Java™ Metadata Interface (JMI) to a metamodel, the apparatus comprising:

means for receiving a JMI implementation request, said request associated with a metamodel, said metamodel comprising at least one package, said at least one package comprising at least one class, said at least one class comprising at least one attribute, reference or operation;

means for implementing a package proxy JMI interface when said request comprises a package proxy request;

means for implementing a class proxy JMI interface when said request comprises a class proxy request; and

means for implementing a class instance JMI interface when said request comprises a class instance request.

68. (NEW) The apparatus of claim 67 wherein said means for implementing a package proxy JMI interface comprises:

means for generating bytecode for a class that implements said package proxy JMI interface; means for creating a new instance of said class; and means for returning said instance.

69. (NEW) The apparatus of claim 68 wherein said means for generating further comprises:

means for receiving a metamodel package;

means for receiving a package proxy interface method associated with said metamodel package;

means for determining a class name based upon said interface method;

means for searching said metamodel package for a class corresponding to said class name;

and

means for producing an implementation of said interface method that returns the proxy for said class when said class name is found in said package.

70. (NEW) The apparatus of claim 69 wherein said implementation of said interface method calls a handler method of the superclass of said class, passing said class name as an argument and returning the proxy for said class.

71. (NEW) The apparatus of claim 67 wherein said means for implementing a class proxy JMI interface comprises:

means for generating bytecode for a class that implements said class proxy JMI interface;

means for creating a new instance of said class; and

means for returning said instance.

72. (NEW) The apparatus of claim 71 wherein said means for generating further comprises:

means for receiving a metamodel class;

means for receiving a class proxy interface method associated with said metamodel class;

means for producing a first implementation of said interface method that creates a new instance of said class when said method is parameterless; and means for producing a second implementation of said interface method that creates a new instance of said class and sets the attributes passed as arguments to said interface method when said interface method includes at least one parameter.

73. (NEW) The apparatus of claim 72 wherein said first implementation calls a handler method of the superclass of said class, passing said class name as an argument and returning a new instance of said class.

74. (NEW) The apparatus of claim 72 wherein said second implementation calls a handler method of the superclass of said class, passing said class name, attributes and attribute values as arguments and returning a new instance of said class.

75. (NEW) The apparatus of claim 67 wherein said means for implementing a class instance JMI interface comprises:

means for generating bytecode for a class that implements said class instance JMI interface; means for creating a new instance of said class; and means for returning said instance.

76. (NEW) The apparatus of claim 75 wherein said means for generating further comprises:

means for receiving a metamodel class;

means for receiving a class instance interface method associated with said metamodel class, said interface method having an interface method name;

means for producing a first implementation of said interface method that sets the value of an attribute when said interface method name includes a first prefix and when the attribute associated with said interface method is found in said metamodel class;

means for producing a second implementation of said interface method that sets the value of a reference when said interface method name includes a first prefix and when the reference associated with said interface method is found in said metamodel class;

means for producing a third implementation of said interface method that gets the value of an attribute when said interface method name includes a second prefix and when the attribute associated with said interface method is found in said metamodel class;

means for producing a fourth implementation of said interface method that gets the value of a reference when said interface method name includes a second prefix and when the reference associated with said interface method is found in said metamodel class; and

means for producing a fifth implementation of said interface method that executes an operation when said interface method has the same name as said operation.

77. (NEW) The apparatus of claim 76 wherein

said first prefix is “set”; and
said second prefix is “get”.

78. (NEW) The apparatus of claim 76 wherein said means for producing a first implementation further comprises:

means for receiving an attribute name and an attribute value; and

means for producing an implementation that calls a handler method of the superclass of said class, passing said attribute name and attribute value as arguments.

79. (NEW) The apparatus of claim 76 wherein said means for producing a second implementation further comprises:

means for receiving a reference name and an reference value; and
means for producing an implementation that calls a handler method of the superclass of said class, passing said reference name and reference value as arguments.

80. (NEW) The apparatus of claim 76 wherein said means for producing a third implementation further comprises:

means for receiving an attribute name;
means for producing an implementation that calls a handler method of the superclass of said class, passing said attribute name as an argument and returning the attribute value associated with said attribute name; and
means for returning said attribute value.

81. (NEW) The apparatus of claim 76 wherein said means for producing a fourth implementation further comprises:

means for receiving a reference name;
means for producing an implementation that calls a handler method of the superclass of said class, passing said reference name as an argument and returning the reference value associated with said reference name; and

means for returning said reference value.

82. (NEW) The apparatus of claim 76 wherein said means for producing a fifth implementation further comprises:

means for receiving an operation name and any associated arguments;

means for producing an implementation that calls a handler method of the superclass of said class, passing said operation name and said associated arguments as arguments and returning an operation return value; and

means for returning said operation return value.

83. (NEW) An apparatus for dynamic implementation of a Java™ Metadata Interface (JMI), the apparatus comprising:

means for receiving a JMI implementation request, said request associated with a metamodel, said metamodel comprising at least one package, said at least one package comprising at least one class, said at least one class comprising at least one attribute, reference or operation;

means for implementing a JMI interface when said JMI interface is unimplemented; and

means for executing a stored JMI interface implementation when said JMI interface is implemented.

84. (NEW) The apparatus of claim 83 wherein

said means for implementing further comprises:

means for implementing a package proxy JMI interface when said request comprises a package proxy request and when said package proxy JMI interface is unimplemented;

means for implementing a class proxy JMI interface when said request comprises a class proxy request and when said class proxy JMI interface is unimplemented; and

means for implementing a class instance JMI interface when said request comprises a class instance request and when said class instance JMI interface is unimplemented;

and

said means for executing further comprises:

means for executing a stored a package proxy JMI interface implementation when said request comprises a package proxy request and when said package proxy JMI interface is implemented;

means for executing a stored class proxy JMI interface when said request comprises a class proxy request and when said class proxy JMI interface is implemented; and

means for executing a stored class instance JMI interface when said request comprises a class instance request and when said class instance JMI interface is implemented.

85. (NEW) The apparatus of claim 84 wherein said means for implementing a package proxy JMI interface comprises:

means for generating bytecode for a class that implements said package proxy JMI interface;

means for creating a new instance of said class; and

means for returning said instance.

86. (NEW) The apparatus of claim 85 wherein said means for generating further comprises:

means for receiving a metamodel package;
means for receiving a package proxy interface method associated with said metamodel package;
means for determining a class name based upon said interface method;
means for searching said metamodel package for a class corresponding to said class name;
and
means for producing an implementation of said interface method that returns the proxy for said class when said class name is found in said package.

87. (NEW) The apparatus of claim 86 wherein said implementation of said interface method calls a handler method of the superclass of said class, passing said class name as an argument and returning the proxy for said class.

88. (NEW) The apparatus of claim 84 wherein said means for implementing a class proxy JMI interface comprises:

means for generating bytecode for a class that implements said class proxy JMI interface;
means for creating a new instance of said class; and
means for returning said instance.

89. (NEW) The apparatus of claim 88 wherein said generating further comprises:

means for receiving a metamodel class;
means for receiving a class proxy interface method associated with said metamodel class;

means for producing a first implementation of said interface method that creates a new instance of said class when said method is parameterless; and means for producing a second implementation of said interface method that creates a new instance of said class and sets the attributes passed as arguments to said interface method when said interface method includes at least one parameter.

90. (NEW) The apparatus of claim 89 wherein said first implementation calls a handler method of the superclass of said class, passing said class name as an argument and returning a new instance of said class.

91. (NEW) The apparatus of claim 89 wherein said second implementation calls a handler method of the superclass of said class, passing said class name, attributes and attribute values as arguments and returning a new instance of said class.

92. (NEW) The apparatus of claim 84 wherein said means for implementing a class instance JMI interface comprises:

means for generating bytecode for a class that implements said class instance JMI interface; means for creating a new instance of said class; and means for returning said instance.

93. (NEW) The apparatus of claim 92 wherein said means for generating further comprises:

means for receiving a metamodel class;

means for receiving a class instance interface method associated with said metamodel class, said interface method having an interface method name;

means for producing a first implementation of said interface method that sets the value of an attribute when said interface method name includes a first prefix and when the attribute associated with said interface method is found in said metamodel class;

means for producing a second implementation of said interface method that sets the value of a reference when said interface method name includes a first prefix and when the reference associated with said interface method is found in said metamodel class;

means for producing a third implementation of said interface method that gets the value of an attribute when said interface method name includes a second prefix and when the attribute associated with said interface method is found in said metamodel class;

means for producing a fourth implementation of said interface method that gets the value of a reference when said interface method name includes a second prefix and when the reference associated with said interface method is found in said metamodel class; and

means for producing a fifth implementation of said interface method that executes an operation when said interface method has the same name as said operation.

94. (NEW) The apparatus of claim 93 wherein

said first prefix is “set”; and

said second prefix is “get”.

95. (NEW) The apparatus of claim 93 wherein said means for producing a first implementation further comprises:

means for receiving an attribute name and an attribute value; and

means for producing an implementation that calls a handler method of the superclass of said class, passing said attribute name and attribute value as arguments.

96. (NEW) The apparatus of claim 93 wherein said means for producing a second implementation further comprises:

means for receiving a reference name and an reference value; and
means for producing an implementation that calls a handler method of the superclass of said class, passing said reference name and reference value as arguments.

97. (NEW) The apparatus of claim 93 wherein said means for producing a third implementation further comprises:

means for receiving an attribute name;
means for producing an implementation that calls a handler method of the superclass of said class, passing said attribute name as an argument and returning the attribute value associated with said attribute name; and
means for returning said attribute value.

98. (NEW) The apparatus of claim 93 wherein said means for producing a fourth implementation further comprises:

means for receiving a reference name;
means for producing an implementation that calls a handler method of the superclass of said class, passing said reference name as an argument and returning the reference value associated with said reference name; and

means for returning said reference value.

99. (NEW) The apparatus of claim 93 wherein said means for producing a fifth implementation further comprises:

means for receiving an operation name and any associated arguments;

means for producing an implementation that calls a handler method of the superclass of said class, passing said operation name and said associated arguments as arguments and returning an operation return value; and

means for returning said operation return value.

100. (NEW) An apparatus for dynamic implementation of a Java™ Metadata Interface (JMI) to a metamodel, the apparatus comprising:

a requestor to make a JMI implementation request, said request associated with a metamodel, said metamodel comprising at least one package, said at least one package comprising at least one class, said at least one class comprising at least one attribute, reference or operation;

a package proxy implementor to implement a package proxy JMI interface when said request comprises a package proxy request;

a class proxy implementor to implement a class proxy JMI interface when said request comprises a class proxy request; and

a class instance implementor to implement a class instance JMI interface when said request comprises a class instance request.

101. (NEW) The apparatus of claim 100 wherein said package proxy implementor is further configured to:

generate bytecode for a class that implements said package proxy JMI interface;
create a new instance of said class; and
return said instance.

102. (NEW) The apparatus of claim 101 wherein said package proxy implementor is further configured to

receive a metamodel package;
receive a package proxy interface method associated with said metamodel package;
determine a class name based upon said interface method;
search said metamodel package for a class corresponding to said class name; and
produce an implementation of said interface method that returns the proxy for said class
when said class name is found in said package.

103. (NEW) The apparatus of claim 102 wherein said implementation of said interface method calls a handler method of the superclass of said class, passing said class name as an argument and returning the proxy for said class.

104. (NEW) The apparatus of claim 100 wherein said class proxy implementor is further configured to:

generate bytecode for a class that implements said class proxy JMI interface;
create a new instance of said class; and

return said instance.

105. (NEW) The apparatus of claim 104 wherein said class proxy implementor is further configured to:

receive a metamodel class;

receive a class proxy interface method associated with said metamodel class;

produce a first implementation of said interface method that creates a new instance of said class when said method is parameterless; and

produce a second implementation of said interface method that creates a new instance of said class and sets the attributes passed as arguments to said interface method when said interface method includes at least one parameter.

106. (NEW) The apparatus of claim 105 wherein said first implementation calls a handler method of the superclass of said class, passing said class name as an argument and returning a new instance of said class.

107. (NEW) The apparatus of claim 105 wherein said second implementation calls a handler method of the superclass of said class, passing said class name, attributes and attribute values as arguments and returning a new instance of said class.

108. (NEW) The apparatus of claim 99 wherein said class instance implementor is further configured to:

generate bytecode for a class that implements said class instance JMI interface;
create a new instance of said class; and
return said instance.

109. (NEW) The apparatus of claim 108 wherein said class instance implementor is further configured to:

receive a metamodel class;
receive a class instance interface method associated with said metamodel class, said interface method having an interface method name;
produce a first implementation of said interface method that sets the value of an attribute when said interface method name includes a first prefix and when the attribute associated with said interface method is found in said metamodel class;
produce a second implementation of said interface method that sets the value of a reference when said interface method name includes a first prefix and when the reference associated with said interface method is found in said metamodel class;
produce a third implementation of said interface method that gets the value of an attribute when said interface method name includes a second prefix and when the attribute associated with said interface method is found in said metamodel class;
produce a fourth implementation of said interface method that gets the value of a reference when said interface method name includes a second prefix and when the reference associated with said interface method is found in said metamodel class; and
produce a fifth implementation of said interface method that executes an operation when said interface method has the same name as said operation.

110. (NEW) The apparatus of claim 109 wherein

said first prefix is “set”; and

said second prefix is “get”.

111. (NEW) An apparatus for dynamic implementation of a Java™ Metadata Interface (JMI), the apparatus comprising:

a requestor to make a JMI implementation request, said request associated with a metamodel,

said metamodel comprising at least one package, said at least one package comprising at least one class, said at least one class comprising at least one attribute, reference or operation;

an implementor to implement a JMI interface when said JMI interface is unimplemented; and an executor to execute a stored JMI interface implementation when said JMI interface is implemented.

112. (NEW) The apparatus of claim 111 wherein

said implementor further comprises:

a package proxy implementor to implement a JMI interface when said request comprises a package proxy request and when said package proxy JMI interface is unimplemented;

a class proxy implementor to implement a JMI interface when said request comprises a class proxy request and when said class proxy JMI interface is unimplemented; and

a class instance implementor to implement a JMI interface when said request comprises a class instance request and when said class instance JMI interface is unimplemented; and
said executor is further configured to:
execute a stored a package proxy JMI interface implementation when said request comprises a package proxy request and when said package proxy JMI interface is implemented;
execute a stored class proxy JMI interface when said request comprises a class proxy request and when said class proxy JMI interface is implemented; and
execute a stored class instance JMI interface when said request comprises a class instance request and when said class instance JMI interface is implemented.

113. (NEW) The apparatus of claim 112 wherein said package proxy implementor is further configured to:

generate bytecode for a class that implements said package proxy JMI interface; create a new instance of said class; and
return said instance.

114. (NEW) The apparatus of claim 113 wherein said package proxy implementor is further configured to:

receive a metamodel package;
receive a package proxy interface method associated with said metamodel package;
determine a class name based upon said interface method;
search said metamodel package for a class corresponding to said class name; and

produce an implementation of said interface method that returns the proxy for said class when said class name is found in said package.

115. (NEW) The apparatus of claim 114 wherein said implementation of said interface method calls a handler method of the superclass of said class, passing said class name as an argument and returning the proxy for said class.

116. (NEW) The apparatus of claim 112 wherein said class proxy implementor is further configured to:

generate bytecode for a class that implements said class proxy JMI interface; create a new instance of said class; and return said instance.

117. (NEW) The apparatus of claim 116 wherein said class proxy implementor is further configured to:

receive a metamodel class; receive a class proxy interface method associated with said metamodel class; produce a first implementation of said interface method that creates a new instance of said class when said method is parameterless; and produce a second implementation of said interface method that creates a new instance of said class and sets the attributes passed as arguments to said interface method when said interface method includes at least one parameter.

118. (NEW) The apparatus of claim 117 wherein said first implementation calls a handler method of the superclass of said class, passing said class name as an argument and returning a new instance of said class.

119. (NEW) The apparatus of claim 117 wherein said second implementation calls a handler method of the superclass of said class, passing said class name, attributes and attribute values as arguments and returning a new instance of said class.

120. (NEW) The apparatus of claim 112 wherein said class instance implementor is further configured to:

- generate bytecode for a class that implements said class instance JMI interface;
- create a new instance of said class; and
- return said instance.

121. (NEW) The apparatus of claim 120 wherein said class instance implementor is further configured to:

- receive a metamodel class;
- receive a class instance interface method associated with said metamodel class, said interface method having an interface method name;
- produce a first implementation of said interface method that sets the value of an attribute when said interface method name includes a first prefix and when the attribute associated with said interface method is found in said metamodel class;

produce a second implementation of said interface method that sets the value of a reference when said interface method name includes a first prefix and when the reference associated with said interface method is found in said metamodel class;

produce a third implementation of said interface method that gets the value of an attribute when said interface method name includes a second prefix and when the attribute associated with said interface method is found in said metamodel class;

produce a fourth implementation of said interface method that gets the value of a reference when said interface method name includes a second prefix and when the reference associated with said interface method is found in said metamodel class; and

produce a fifth implementation of said interface method that executes an operation when said interface method has the same name as said operation.

122. (NEW) The apparatus of claim 121 wherein

 said first prefix is “set”; and
 said second prefix is “get”.